EUROPEAN PATENT OFFICE

Patent Abstracts of Japan

PUBLICATION NUMBER : 2000132671
PUBLICATION DATE : 12-05-00

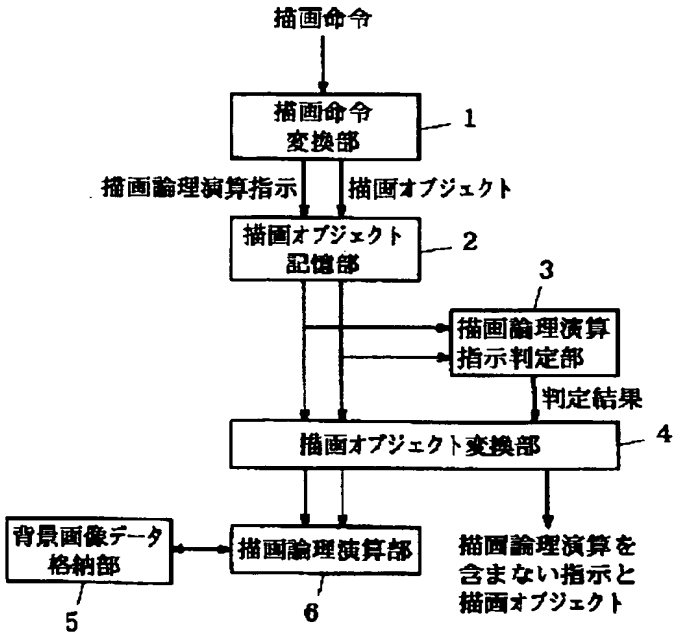
APPLICATION DATE : 04-12-98
APPLICATION NUMBER : 10345458

APPLICANT : FUJI XEROX CO LTD;

INVENTOR : SEKINE HIROSHI;

INT.CL. : G06T 1/00 H04N 1/387

TITLE : IMAGE PROCESSOR AND IMAGE PROCESSING METHOD



ABSTRACT : PROBLEM TO BE SOLVED: To provide an image processor and an image processing method by which a plotting processing can be executed at a high speed.

SOLUTION: When a plotting instruction including a plotting arithmetic instruction is inputted to a plotting instruction converting part 1, the part 1 converts the inputted plotting instruction into a plotting object and the plotting arithmetic indication and stores them in a plotting object storage part 2. A plotting arithmetic instruction judging part 3 judges whether or not the plotting arithmetic instruction stored in the part 2 is provided with instruction contents to be plotted without executing a plotting arithmetic processing. When it is judged that the instruction contents can be processed without the plotting arithmetic processing as the result of judgement, a plotting object converting part 4 converts the judged plotting arithmetic instruction and the plotting object into the ones without requiring a plotting logical calculation and the processing after that is continued by another processing part.

COPYRIGHT: (C)2000,JPO

(19)日本国特許庁（ＪＰ）　(12) **公開特許公報（Ａ）**　(11)特許出願公開番号
特開2000－132671
（P2000－132671A）
(43)公開日　平成12年５月12日(2000.5.12)

(51)Int.Cl.'	識別記号	Ｆ Ｉ		テーマコード'(参考)
Ｇ０６Ｔ　1/00		Ｃ０６Ｆ　15/66	４５０	５Ｂ０５７
Ｈ０４Ｎ　1/387		Ｈ０４Ｎ　1/387		５Ｃ０７６

審査請求　有　請求項の数９　ＯＬ　（全 16 頁）

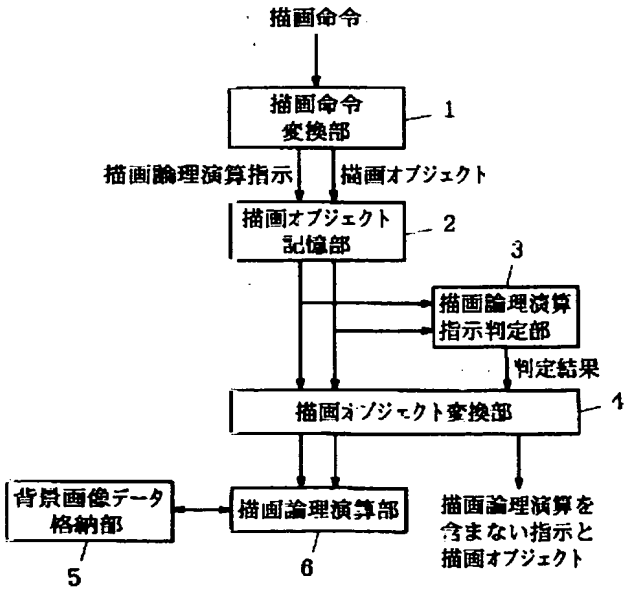
(21)出願番号　特願平10－345458
(22)出願日　平成10年12月４日(1998.12.4)
(31)優先権主張番号　特願平10－231759
(32)優先日　平成10年８月18日(1998.8.18)
(33)優先権主張国　日本（ＪＰ）

(71)出願人　000005496
富士ゼロックス株式会社
東京都港区赤坂二丁目17番22号
(72)発明者　國政　武史
神奈川県海老名市本郷2274番地　富士ゼロックス株式会社内
(72)発明者　柴田　文彦
神奈川県海老名市本郷2274番地　富士ゼロックス株式会社内
(74)代理人　100101948
弁理士　柳澤　正夫

最終頁に続く

(54)【発明の名称】　画像処理装置および画像処理方法

(57)【要約】
【課題】　高速な描画処理を可能とした画像処理装置および画像処理方法を提供する。
【解決手段】　描画命令変換部１に描画論理演算指示を含む描画命令が入力されると、描画命令変換部１は入力された描画命令を描画オブジェクトと描画論理演算指示に変換し、描画オブジェクト記憶部２に格納する。描画論理演算指示判定部３は、描画オブジェクト記憶部２に格納されている描画論理演算指示が、描画論理演算処理を行わなくても描画できる指示内容であるか否かを判定する。判定の結果、描画論理演算処理を行わなくても処理可能な指示内容であると判断された場合には、描画オブジェクト変換部４によって、判定した描画論理演算指示および描画オブジェクトを、描画論理演算を必要としない指示および描画オブジェクトに変換し、他の処理部にその後の処理を継続する。



BNSDOCID: <JP_____2000132671A__I_>

【特許請求の範囲】

【請求項１】　画像を構成する画素間の論理演算処理の指示を含む描画命令を受け取って処理を行う画像処理装置において、前記描画命令を描画オブジェクトおよび描画論理演算指示に変換する描画命令変換手段と、前記描画論理演算指示が論理演算処理を行わなくても処理可能な指示内容であるか否かを判定する描画論理演算指示判定手段と、前記描画論理演算指示判定手段において前記描画論理演算指示が論理演算処理を行わなくても処理可能な指示内容であると判定された場合に前記描画オブジェクトおよび描画論理演算指示を描画論理演算を必要としない指示および描画オブジェクトに変換する描画オブジェクト変換手段を有することを特徴とする画像処理装置。

【請求項２】　画像を構成する画素間の論理演算処理の指示を含む描画命令を受け取って処理を行う画像処理装置において、前記描画命令を描画オブジェクトおよび描画論理演算指示に変換する描画命令変換手段と、前記描画命令変換手段で変換された前記描画オブジェクトおよび前記論理演算指示を記憶する描画オブジェクト記憶手段と、前記描画オブジェクト記憶手段に記憶されている連続した複数の前記描画論理演算指示が論理演算処理を行わなくても処理可能な指示内容であるか否かを判定する描画論理演算指示判定手段と、前記描画論理演算指示判定手段において前記連続した複数の描画論理演算指示が論理演算処理を行わなくても処理可能な指示内容であると判定された場合に前記連続した複数の描画論理演算指示および前記描画オブジェクトを描画論理演算を必要としない指示および描画オブジェクトに変換する描画オブジェクト変換手段を有することを特徴とする画像処理装置。

【請求項３】　さらに、背景画像データを格納する背景画像データ格納手段と、前記背景画像データ格納手段に格納されている前記背景画像データと前記描画オブジェクトとの論理演算処理を前記描画論理演算指示に従って行い結果データを前記背景画像データ格納手段に書き戻す描画論理演算手段を有することを特徴とする請求項１または請求項２に記載の画像処理装置。

【請求項４】　前記描画オブジェクトは、形状を定義するマスクパターン、塗りつぶしパターンを定義する背景パターンのうちの少なくとも１つからなり、前記描画論理演算手段は、前記マスクパターン、前記背景パターン、および前記背景画像データのうちの指定された２つのデータに対して論理積、論理和、あるいは排他的論理和のうちのいずれかの演算を行うことを特徴とする請求項３に記載の画像処理装置。

【請求項５】　前記描画オブジェクトは、形状を定義するマスクパターン、塗りつぶしパターンを定義する背景パターンのうちの少なくとも１つからなり、前記描画論理演算手段は、前記マスクパターン、前記背景パターン、および前記背景画像データのうちの指定された２つのデータに対して論理積、論理和、あるいは排他的論理和の演算を組み合わせて行うことを特徴とする請求項３に記載の画像処理装置。

【請求項６】　前記描画論理演算指示判定手段は、特定の描画論理演算指示を検出するとともに、形状を定義するマスクパターンまたは塗りつぶしパターンを定義する背景パターンが特定の条件に一致するか否かを判定することを特徴とする請求項１または請求項２に記載の画像処理装置。

【請求項７】　画像を構成する画素間の論理演算処理の指示を含む描画命令を受け取って処理を行う画像処理方法において、前記描画命令を描画オブジェクトおよび描画論理演算指示に変換し、前記描画論理演算指示が論理演算処理を行わなくても処理可能な指示内容であるか否かを判定し、前記描画論理演算指示が論理演算処理を行わなくても処理可能な指示内容であると判定された場合には、前記描画オブジェクトおよび描画論理演算指示を描画論理演算を必要としない指示および描画オブジェクトに変換することを特徴とする画像処理方法。

【請求項８】　画像を構成する画素間の論理演算処理の指示を含む描画命令を受け取って処理を行う画像処理方法において、前記描画命令を描画オブジェクトおよび描画論理演算指示に変換し、連続した複数の前記描画論理演算指示が論理演算処理を行わなくても処理可能な指示内容であるか否かを判定し、前記連続した複数の描画論理演算指示が論理演算処理を行わなくても処理可能な指示内容であると判定された場合に、前記連続した複数の描画論理演算指示および前記描画オブジェクトを描画論理演算を必要としない指示および描画オブジェクトに変換することを特徴とする画像処理方法。

【請求項９】　前記描画論理演算指示が論理演算処理を行わなくても処理可能な指示内容であるか否かの判定は、前記描画論理演算指示が特定の描画論理演算指示であると判定された場合に、形状を定義するマスクパターンまたは塗りつぶしパターンを定義する背景パターンが特定の条件に一致するか否かを判定するものであることを特徴とする請求項７または請求項８に記載の画像処理方法。

【発明の詳細な説明】

【０００１】

【発明の属する技術分野】本発明は、例えば画像作成装置などから出力される描画命令、特に画像を構成する画素間の論理演算処理の指示を含む描画命令を受け取って処理を行う画像処理装置および画像処理方法に関するものである。

【０００２】

【従来の技術】パーソナルコンピュータなどの画像作成装置などから出力される様々な描画命令を処理し、印刷装置で印刷可能な画素の並びで表現された画像データに

展開する画像形成システムにおいては、描画命令を受け取ってから画像データに展開されるまでの間に描画命令に応じた様々な処理が行われる。その一つとして、複数の画像が重なる位置に描かれる場合に、もとの画像と、新たに書き込む画像との重ね合わせの方法の指示に従って、両方の画像を重ね合わせる処理がある。指示される重ね合わせの方法は、通常、論理和、論理積、排他的論理和などの論理演算の種類として指示される。このような画像間の論理演算は、一般にＲＯＰ（Ｒａｓｔｅｒ ＯＰｅｒａｔｉｏｎ）として知られている。

【０００３】ＲＯＰ処理は、一般に複雑で処理時間がかかり、実行のためのメモリも多く必要とするものが多い。特に、１つの効果を表現するために、複数のＲＯＰ処理を組み合わせたコマンドを出力するようなアプリケーションプログラムも数多く存在する。このようなＲＯＰ処理を組み合わせた指示が、印刷装置によって画像が印刷されるまでの処理時間を増大させる要因となっていた。

【０００４】ＲＯＰ処理の種類としては、通常、１６ないし２５６種類が用意されている。しかしながら、これらのＲＯＰ処理のすべてが常用されるわけではなく、アプリケーションプログラムによって、限られた種類のものを特徴的な組み合わせパターンによって使用する場合が多い。

【０００５】ＲＯＰ処理の組み合わせの例としては、３ステップのＲＯＰによって、任意形状の内側をグラデーションで塗りつぶすような処理を挙げることができる。この処理のステップ１では、任意形状を含む矩形領域の元からあった画素と、グラデーションの塗り潰しパターンとを排他的論理和（ＸＯＲ）のＲＯＰ処理で演算し、目的領域の画素として描画する。次のステップ２では、任意形状を表す白黒のマスクパターンと、目的の領域の画素とを論理積（ＡＮＤ）のＲＯＰ処理で演算し、目的の領域の画素として描画する。この結果、任意形状の内側はマスクパターンの黒画素で置き換えられ、外側は白画素と論理和を取った結果、元の画素値のまま残ることになる。最後のステップ３では、ステップ１と同じ矩形領域の画素と、前述のグラデーションの塗り潰しパターンとを、再度排他的論理和（ＸＯＲ）のＲＯＰ処理で演算し、目的の領域の画素として描画する。この結果、ステップ２で黒く塗り潰された部分、つまり任意形状の内側部分は、当初の目的であるグラデーションパターンで塗り潰され、その外側部分は、この３ステップのＲＯＰ処理を始める以前の状態に戻ることになる。

【０００６】上述のような任意形状の内側を特定のパターンで塗りつぶす処理を行うためには、３回のＲＯＰ処理を行わなければならない。そのため、処理に時間がかかってしまう。そこで、例えば特開平８－２６３６７４号公報にも記載されているように、このようなＲＯＰ処理の特徴的パターンを検出して、ＲＯＰ処理のステップ数を削減し、より単純なＲＯＰ処理に置き換えようという技術が考えられている。この文献に開示されている方法では、３ステップのＲＯＰ処理のうち、最初のステップを完全に省き、ステップ２のマスクパターンと、ステップ３のグラデーションパターンとの論理積をあらかじめ演算した上で、目的の領域に書き込む。これによって、ＲＯＰ処理の実行回数を少なくし、高速な処理を実現している。

【０００７】上述の例では３ステップのＲＯＰ処理のうちの１ステップを割愛することによって高速処理を実現しているが、ＲＯＰ処理を実行することには変わりがない。また、他のＲＯＰ処理の組み合わせパターンについては記載されておらず、他のＲＯＰ処理の組み合わせについても高速化が期待されている。

【０００８】一方、上述のようなＲＯＰ処理は、もともとパーソナルコンピュータ等のディスプレイ装置上の表示を対象に考え出されたものであり、論理演算の対象となる画素データは、ＲＧＢ色空間で表現されていることを前提としている。一般の印刷装置では、最終的にＣＭＹＫ色空間を使用しており、ＲＯＰ処理が前提としている色空間とは異なっている。ＲＯＰ処理は、ＣＭＹＫ色空間のようにＫ色成分が他のＣ，Ｍ，Ｙ色成分に依存するような一次独立でない色空間では正しく実行することができない。そのため、ＣＭＹＫ色空間で表現された画素に対してＲＯＰ処理を実行すると、ＲＧＢ色空間の画素に対して実行した場合とは異なる結果になる。

【０００９】このため、ＲＯＰ処理を正しく実行するための一つの方法として、例えば、ＲＧＢ色空間で表現された画像データを、はじめにＲＧＢ色空間のまま画素データに展開しながらＲＯＰ処理を施す。そして、１ページ内の画像データをすべて画素データに展開処理し、ＲＯＰ処理も終わった後に、１ページ分すべての画素をまとめてＣＭＹＫ色空間に変換するという方法が考えられる。しかし、この方法では、ＲＧＢ色空間で表現された画素データをすべて画素単位でＣＭＹＫ色空間の画素に色変換する必要がある。その結果、色変換に多大な処理時間を要することになる。さらに、元の描画コマンドが表す画像データの種類によらず、一様の色変換を施すことになるので、画像データの種類に適した色変換パラメータを選ぶことができず、画質上の問題も発生することになる。

【００１０】このような方法に対して、いったんＣＭＹＫ色空間の画素データに展開した画像データを、ＲＯＰ処理が必要な部分だけＣＭＹ色空間あるいはＲＧＢ色空間に変換してからＲＯＰ処理を施し、その後ＣＭＹ色空間またはＲＧＢ色空間に変換した部分だけを再びＣＭＹＫ色空間に変換するという方法も考えられる。この方法によれば、先の方法のようにＲＧＢ空間でいったん展開し、展開したすべての画素をそれぞれＣＭＹＫ色空間に変換する場合に比べ、高速な処理が期待できる。また、

BNSDOCID: <JP____2000132671A__I_>

CMYK色空間のまま処理する場合と比較すれば、正しいROP処理が可能となる。しかし、CMYK色空間からCMY色空間あるいはRGB色空間への変換は1対1の変換ではなく、変換後の色値が一意に決まらないため、ROP処理のためにCMY色空間またはRGB色空間に変換した部分を再びCMYK色空間に変換しても、画素値が元に戻るとは限らない。そのため、ROP処理後にCMYK色空間に再変換したときに、ROP処理した部分とその周囲のROP処理をしなかった部分との間に色の段差ができてしまう場合がある。

【0011】このように、ROP処理を行うことによってROP処理に伴う色変換処理により色再現上の不具合が発生する場合がある。このような不具合を回避するためにも、ROP処理を少なくすることが望まれる。

【0012】

【発明が解決しようとする課題】本発明は、上述した事情に鑑みてなされたもので、高速および高画質の描画処理を可能とした画像処理装置および画像処理方法を提供することを目的とするものである。

【0013】

【課題を解決するための手段】本発明は、描画命令中の描画論理演算指示が論理演算処理を行わなくても処理可能な指示内容であるか否かを判定し、論理演算処理を行わなくても処理可能な指示内容である1個あるいは連続した複数の描画論理演算指示および描画オブジェクトを、描画論理演算を必要としない指示および描画オブジェクトに変換することを特徴とするものである。これによって、必要のない描画論理演算を行わなくて済むので、高速および高画質の描画処理が可能となる。

【0014】

【発明の実施の形態】図1は、本発明の実施の一形態を示すブロック図である。図中、1は描画命令変換部、2は描画オブジェクト記憶部、3は描画論理演算指示判定部、4は描画オブジェクト変換部、5は背景画像データ格納部、6は描画論理演算部である。描画命令変換部1は、画像を構成する画素間の論理演算処理の指示を含む描画命令を受け取って、描画オブジェクトと描画論理演算指示に変換する。描画オブジェクトは、形状を定義するマスクパターン、塗りつぶしパターンを定義する背景パターンの少なくとも1つ以上から構成される。また、論理演算指示としては、従来より知られているROP処理のいずれか指定される。

【0015】描画オブジェクト記憶部2は、描画命令変換部1で変換した描画オブジェクトおよび描画論理演算指示を記憶する。なお、この描画オブジェクト記憶部2を配置しない構成も可能であるが、後述するように描画論理演算指示判定部3で連続する複数の描画論理演算指示について判定を行う際には、この描画オブジェクト記憶部2に複数の描画論理演算指示および対応する描画オブジェクトを記憶させておくと便利である。

【0016】描画論理演算指示判定部3は、描画論理演算指示を解析し、その描画論理演算指示あるいは連続した複数の描画論理演算指示が、論理演算処理を行わなくても処理可能な指示内容であるか否かを判定する。この判定は、例えば描画論理演算指示が特定の描画論理演算指示である場合や、連続する複数の描画論理演算指示が特定の描画論理演算指示のパターンであることを条件とすることができる。また、この判定の際に、その描画論理演算指示によって指示されている論理演算処理の対象となる描画オブジェクトを参照して判定することもできる。

【0017】描画オブジェクト変換部4は、描画論理演算指示判定部3において論理演算処理を行わなくても処理可能な指示内容であると判定された1ないし複数の論理演算指示、その論理演算指示に対応する描画オブジェクトについて、描画論理演算を必要としない指示および描画オブジェクトに変換する。

【0018】背景画像データ格納部5は、描画論理演算部6で用いられる背景画像データを格納するとともに、描画論理演算部6で演算された結果の画像データを格納する。

【0019】描画論理演算部6は、描画オブジェクト変換部4で論理演算処理を行わなくても処理可能な指示内容であると判定された以外の描画論理演算指示について、その描画論理演算指示に従って、描画オブジェクトおよび背景画像データ格納部5に格納されている背景画像データ間の論理演算処理を実行し、その結果を背景画像データ格納部5に格納する。

【0020】上述の本発明の実施の一形態における動作の概要を説明する。描画命令変換部1に描画論理演算指示を含む描画命令が入力されると、描画命令変換部1は入力された描画命令を描画オブジェクトと描画論理演算指示に変換し、描画オブジェクト記憶部2に格納する。描画論理演算指示判定部3は、描画オブジェクト記憶部2に格納されている1つの描画論理演算指示あるいは連続する複数の描画論理演算指示が、描画論理演算処理を行わなくても描画できる指示内容であるか否かを判定する。判定の結果、描画論理演算処理を行わなくても処理可能な指示内容であると判断された場合には、描画オブジェクト変換部4によって、判定した描画論理演算指示および描画オブジェクトを、描画論理演算を必要としない指示および描画オブジェクトに変換し、他の処理部にその後の処理を継続する。

【0021】一方、描画論理演算指示判定部3によって描画論理演算処理が必要な指示内容であると判断された場合には、描画論理演算部6は、背景画像データ格納部5から背景画像データを読み出し、描画論理演算指示に従い、読み出した背景画像データと描画オブジェクトとの描画論理演算を行う。この演算が行われた結果の画像データは、背景画像データ格納部5に書き戻される。

BNSDOCID: <JP____2000132671A__I_>

【００２２】図２は、本発明の実施の一形態を実現する画像形成システムの一例を示すブロック図である。図中、１１はホストコンピュータ、１２はアプリケーションプログラム、１３はプリンタドライバ、１４はオペレーティングシステム、１５はハードウェア、１６は入力装置、１７は表示装置、１８は外部記憶装置、１９は伝送路、２０はプリンタ、２１はインタープリタ部、２２はイメージャ部、２３はレンダラ部、２４はページメモリ、２５はプリンタエンジン、２６は制御ハードウェアである。この画像形成システムでは、ホストコンピュータ１１において取得あるいは生成した画像を、プリンタ２０において被記録媒体上に形成するものである。

【００２３】ホストコンピュータ１１は、ＣＰＵやメモリ等のハードウェア１５を内蔵し、また外部機器としてキーボードやマウスなどの入力装置１６、ＣＲＴなどの表示装置１７、ハードディスクやＣＤ－ＲＯＭなどの外部記憶装置１８等が接続されている。さらに、インタフェースを介して伝送路１９によりプリンタ２０と接続されている。これらのハードウェア１５や各種の装置、および、各種のソフトウェアは、オペレーティングシステム１４によって制御、管理されている。このオペレーティングシステム１４による制御および管理のもとで、アプリケーションプログラム１２が動作する。アプリケーションプログラム１２は、文字や画像などを印刷する指示を行うことができる。ここでは、印刷する内容はページ記述言語などによって記述された描画情報として出力されるものとする。

【００２４】プリンタドライバ１３は、プリンタ２０に付属して提供されるものであり、ホストコンピュータ１１にロードされて機能するプログラムである。通常、プリンタドライバ１３は、特定のプリンタ２０での使用が予想される様々なデータ形式にホストコンピュータ１１を対応させるために使用される。プリンタドライバ１３は、アプリケーションプログラム１２から描画情報を受け取ると、プリンタ２０に送信する前に、様々な処理ステップを実行する。例えば、アプリケーションプログラム１２から受け取った描画情報をプリンタ２０で解釈可能なコマンドおよび描画オブジェクトの列に変換して送信することができる。あるいは、描画情報のデータ形式のままプリンタ２０へ送信してもよい。

【００２５】プリンタ２０では、ホストコンピュータ１１から伝送路１９を介して伝送されてくる描画情報あるいはコマンドと描画データの列などを受信し、被記録媒体上に画像を形成する。そのために、制御ハードウェア２６，ページメモリ２４，プリンタエンジン２５などのハードウェアと、制御ハードウェア２６上で動作するインタープリタ部２１，イメージャ部２２，レンダラ部２３などのソフトウェアを有している。

【００２６】インタープリタ部２１は、プリンタ２０が受信した描画情報やコマンドと描画データの列などを認識し、コマンドと引数に組み立ててコマンドを解釈する。そして、解釈したコマンドの内容に従って、イメージャ部２２を呼び出してグラフィックス描画、文字描画、イメージ処理を行う。イメージャ部２２は、インタープリタ部２１の指示に従って画像を描画する。レンダラ部２３は、イメージャ部２２で描画された画像をラスタ化し、ページメモリ２４に記憶させる。このとき、例えばカラー印刷を行うのであれば、それぞれの色ごとに、例えばＣＭＹＫの４プレーンにラスタ化し、ページメモリ２４に記憶させる。プリンタエンジン２５は、ページメモリ２４に格納された画像データに従って実際に被記録媒体上に画像を形成する。プリンタエンジン２５としては、例えばレーザ方式を始め、種々の方式の記録方法を採用することができる。

【００２７】このような画像形成システムにおいて、ホストコンピュータ１１に導入されているプリンタドライバ１３によって、図１に示した本発明の画像処理方法の全部の処理または描画オブジェクト変換部４までの処理を実行することができる。すなわち、アプリケーションプログラム１２から出力される描画情報を参照し、そのうちの１つの描画論理演算指示あるいは連続した複数の描画論理演算指示が、画素間の描画論理演算を必要としない場合、その描画論理演算指示および描画オブジェクトを、描画論理演算を行わないコマンドおよび描画オブジェクトに変換してプリンタ２０に送信する。この場合、描画論理演算部６および背景画像データ格納部５は、ホストコンピュータ１１またはプリンタ２０のいずれに設けてもよい。なお、描画オブジェクト記憶部２を設ける場合には、例えばホストコンピュータ１１のハードウェア１５にあるＲＡＭや外部記憶装置１８の記憶領域の一部を用いて実現することができる。また、背景画像データ格納部５をホストコンピュータ１１側に設ける場合も同様である。

【００２８】あるいは、プリンタ２０に図１に示した本発明の画像処理装置を配置し、あるいはプリンタ２０における処理として図１に示した本発明の画像処理方法を実行させることができる。この場合、プリンタドライバ１３は従来と同様の処理を行えばよい。ホストコンピュータ１１から送られてくる描画情報はプリンタ２０における処理として、描画情報の中の１つの描画論理演算指示あるいは連続した複数の描画論理演算指示が画素間の描画論理演算を必要としない場合、その描画論理演算指示および描画オブジェクトを、描画論理演算を行わないコマンドおよび描画オブジェクトに変換すればよい。これによって、描画論理演算処理を用いずに描画オブジェクトを描画することができる。

【００２９】いずれの場合にも、本発明の処理によって描画論理演算処理を減少させることができ、高速な画像形成を可能とすることができる。また、描画論理演算処理を行わないことによって、描画論理演算処理によって

BNSDOCID: <JP____2000132671A__I_>

生じる画質の劣化、例えば色の意図しない変化などを回避することができる。

【００３０】以下、具体例を用いながら本発明の実施の一形態の動作についてさらに説明する。以下の具体例では、背景画像データ格納部５に格納されている任意のカラー画像Ｄ（以下、ディスティネーション画像と呼ぶ）と、そこに描画する描画オブジェクトであるビットマップ画像Ｓ（以下、ソース画像と呼ぶ）、そして描画オブジェクトの塗り潰しパターンを定義するパターン画像Ｐが仮定される。このうち、ソース画像Ｓはマスクパターン、パターン画像Ｐは背景パターンとも呼ぶことがある。

【００３１】ユーザは、例えばアプリケーションプログラム１２のグラフィカルユーザインタフェース（ＧＵＩ）を使って、これらの画像間のビットマップデータに対して、論理積（ＡＮＤ：ａ）、論理和（ＯＲ：ｏ）、排他的論理和（ＸＯＲ：ｘ）、反転（ＮＯＴ：ｎ）等の論理演算を指定することが可能である。また、これらの論理演算を組み合わせて目的とする描画結果を得ることもできる。所望の描画結果を得るための論理演算の指定は、ユーザの直接的なオペレーションの代わりにアプリケーションプログラム１２によって自動的に行うこともできる。

【００３２】論理演算の種類は、逆ポーランド形式で表す場合、例えば、ＤＰａは、ディスティネーション画像Ｄとパターン画像Ｐとの論理積（ａ）を演算することを表している。その結果は背景画像データ格納部５に書き戻され、新たなディスティネーション画像Ｄとなり得る。

【００３３】また、以下の第１および第２の具体例では、図１に示すような構成の本発明の画像処理方法をプリンタドライバ１３に組み込んだ例について説明する。ユーザは、アプリケーションプログラム１２において、例えばグラフィカルユーザインタフェース（ＧＵＩ）を使用して任意に文字、グラフィックス、イメージ等の描画オブジェクトを画面上に配置してゆく。これらの描画オブジェクトは、重なりを持たせたり、透明度を指定して半透明状態を指定するなど、種々の指定を行うことができる。

【００３４】これらの描画オブジェクトを印刷する場合、アプリケーションプログラム１２は、オペレーティングシステム１４を介してプリンタドライバ１３に対して描画情報を出力する。例えば描画情報は、描画関数と色指定コード、論理演算コード、クリップ領域指定等で構成される。さらに具体的には、文字を示す描画情報は、文字であることを示す描画関数と文字の種類を示す文字コード、文字の色を示す色情報や、描画位置を示す座標情報等からなる。グラフィックスを示す描画情報は、図形の種類を示す描画関数と、論理演算コード、描画するペンの種類や、塗りつぶしのパターン指定、色情報、描画位置を示す座標情報等からなる。また、写真等のイメージを示す描画情報は、ビットマップ情報と、論理演算コード、ビットマップの形式や、描画位置を示す座標情報等からなる。

【００３５】プリンタドライバ１３は、受信した描画情報から、対象とするプリンタ２０に適合したコマンドおよび描画オブジェクトを生成する。このとき、描画論理演算コードを含む描画情報については、本発明の画像処理方法を適用して、描画論理演算が不要な描画論理演算コードを含む描画情報を、描画論理演算を行わないコマンドおよび描画オブジェクトに変換する。

【００３６】第１の具体例では、上述の逆ポーランド形式の記法でＰＳＤＰｘａｘという描画論理演算を考える。図３は、描画論理演算ＰＳＤＰｘａｘの一例の説明図である。この描画論理演算ＰＳＤＰｘａｘは、指定されたソース画像Ｓが白黒イメージで表現された画像である場合、黒い部分はディスティネーション画像Ｄを透過し、白い部分は同時に指定されているパターン画像Ｐで塗りつぶす効果を狙った処理であることが知られている。図３に示す例では、矩形の画像を有するパターン画像Ｐと、パターン画像Ｐの矩形と重なりを有する位置に矩形の画像を有するディスティネーション画像Ｄが存在し、両画像の矩形の重なる部分に市松模様を有する２値画像であるソース画像Ｓが存在する。このとき、描画論理演算ＰＳＤＰｘａｘによってソース画像Ｓが黒い部分はディスティネーション画像Ｄが、白い部分はパターン画像Ｐが選択され、重なり部分で市松模様となった画像Ｄ’が得られる。

【００３７】しかし、この描画論理演算ＰＳＤＰｘａｘでは、ソース画像Ｓが２値画像であればマスキングパターン処理によって代替可能である。図４は、マスキングパターン処理の一例の説明図である。マスキングパターン処理は、マスキングパターンＭの値に応じてパターン画像Ｐをディスティネーション画像Ｄに貼り付けてゆく処理であり、論理演算処理は行わない。このマスキングパターン処理による結果は画像Ｄ’のようになる。

【００３８】このマスキングパターン処理の結果である画像Ｄ’と、図３における描画論理演算処理の結果であるディスティネーション画像Ｄ’とを比較してわかるように、同じ結果が得られる。すなわち、描画論理演算ＰＳＤＰｘａｘを指示する描画論演算指示は、ソース画像Ｓが白黒２値画像であった場合には、描画論理演算を行わなくても処理可能な指示内容である。さらに、この描画論理演算指示は、マスキングパターン処理に変更可能であることがわかる。本発明では、描画論理演算指示がＰＳＤＰｘａｘであり、ソース画像Ｓが白黒２値画像である場合に、その論理演算指示およびソース画像Ｓをマスキングパターン処理を行う指示およびマスキングパターンＭに変換して出力する。これによって、時間のかかる描画論理演算ＰＳＤＰｘａｘを行わずに処理可能とな

BNSDOCID: <JP____2000132671A__I_>

る。また、この描画論理演算ＰＳＤＰｘａｘを行わないことによって、画質劣化の可能性を回避することができる。

【００３９】図５は、本発明の実施の一形態における第１の具体的な変換処理の一例を示すフローチャートである。Ｓ３１において描画論理演算指示を含む描画情報を取得し、その中の描画論理演算コードをＳ３２において調べる。例えば上述の描画論理演算ＰＳＤＰｘａｘが描画論理演算コード１８４であるとすれば、描画論理演算コードが１８４か否かを調べる。描画論理演算コードが１８４であった場合には、さらにＳ３３において、この描画情報と一緒に入力されてきたソース画像Ｓを解析し、ソース画像Ｓが白黒２値画像であるか否かを判定する。解析の結果、ソース画像Ｓが白黒２値画像であると判断された場合、描画オブジェクトおよび描画論理演算指示を描画論理演算を必要としない指示および描画オブジェクトに変換する。すなわちＳ３４において、ソース画像Ｓからマスキングパターンｍを生成するとともに、パターン画像Ｐを使ったマスキングパターン処理を行う命令を生成し、プリンタ２０に対して送信する。

【００４０】一方、描画情報中の描画論理演算コードが１８４でなかった場合、および、描画論理演算コードが１８４であってもソース画像Ｓが白黒２値画像でなかった場合には、Ｓ３５において従来と同様に処理し、例えば論理演算コード指定コマンド、パターン画像指定コマンド、ソース画像指定コマンドとしてプリンタ２０に対して送信する。これらのコマンドを受信したプリンタ２０側では、コマンド指示に従って、受信したソース画像、パターン画像と、ページメモリに記憶されているディスティネーション画像データ間の論理演算を実行してラスターデータを生成することになる。

【００４１】次に第２の具体例として、別の描画論理演算について考える。ここでは、ある描画目的のために複数の描画情報を組み合わせることによって実現する場合について示す。図６は、複数の描画論理演算の組み合わせの一例の説明図である。この例では、第１の描画情報で、第１のパターン画像Ｐとディスティネーション画像Ｄの排他的論理和（ｘ）を指定して、新たなディスティネーション画像Ｄ’を生成する。すなわち上述の逆ポーランド形式の記法でＤＰｘという描画論理演算である。

【００４２】次に第２の描画情報で、白黒の第２のパターン画像Ｐ’と第１の描画情報によって生成したディスティネーション画像Ｄ’との論理積（ａ）を指定して、さらに新しいディスティネーション画像Ｄ”を生成する。これはＤＰａという描画論理演算である。図６では、第２のパターン画像Ｐ’とディスティネーション画像Ｄ’を用いた論理演算であることを明示するため、Ｄ’Ｐ’ａと示している。

【００４３】最後に第３の描画情報で、第１の描画情報で指定したのと同じ第１のパターン画像Ｐと第２の描画情報によって生成したディスティネーション画像Ｄ”との排他的論理和（ｘ）を指定することによって、目的のディスティネーション画像Ｄ’’’を生成する。この描画論理演算は、第１の描画情報の場合と同様のＤＰｘである。図６では、第１のパターン画像Ｐとディスティネーション画像Ｄ”を用いた論理演算であることを明示するため、Ｄ”Ｐａとして示している。

【００４４】このようにして３つの描画情報に含まれる描画論理演算指示によって行われた描画論理演算によって、第２のパターン画像Ｐ’である白黒イメージで表現された画像に対して、黒い部分はディスティネーション画像を透過にし、白い部分は第１と第３の描画情報で指定されているパターン画像（Ｐ）で塗りつぶした画像が得られる。

【００４５】このような３段階の描画論理演算処理によって行われた処理結果は、上述の図４に示したマスキングパターン処理に他ならない。すなわち、第２の描画情報の持つ白黒２値画像のパターン画像Ｐ’をマスキングパターンｍとして、黒い部分は、ディスティネーション画像Ｄを透過にし、白い部分は第１および第３の描画情報で指定した第１のパターン画像Ｐで塗りつぶす効果を狙った処理である。

【００４６】このように、連続する３つの描画情報中の描画論理演算指示がＤＰｘ、ＤＰａ、ＤＰｘの順で並び、第１の描画情報および第３の描画情報における第１のパターン画像Ｐが同一であり、かつ、第２のパターン画像Ｐ’が白黒２値画像である場合には、マスキングパターン処理で代替可能であることがわかる。本発明では、連続する３つの描画情報について、上述のような条件に合致するか否かを判定し、上述の条件に合致する場合にはマスキングパターン処理に変換することによって描画論理演算処理を減らすことができる。

【００４７】図７は、本発明の実施の一形態における第２の具体的な変換処理の一例を示すフローチャートである。なお、ここでは描画論理演算ＤＰｘを指示する論理演算コードが９０．描画論理演算ＤＰａを指示する論理演算コードが１６０であるものとして説明する。まずＳ４１において、第１の描画情報を取得し、Ｓ４２においてその第１の描画情報中の論理演算コードが９０であるか否かを判定する。論理演算コードが９０であると判定された場合には、Ｓ４３において、この第１の描画情報と一緒に入力されてきたパターン画像Ｐとその他の描画情報を例えば描画オブジェクト記憶部２などに保存しておく。これでＳ４１で取得した描画情報に対する処理は終了する。従って、この時点においてはこの第１の描画情報に対するコマンドは出力されない。

【００４８】一方、Ｓ４１で取得した第１の描画情報中の論理演算コードが９０でない場合には、Ｓ５２において、従来と同様に、取得した第１の描画情報についてのコマンドおよび描画オブジェクトを出力して、この第１

BNSDOCID: <JP____2000132671A__I_>

の描画情報についての処理を終了する。

【００４９】次にＳ４４において、第２の描画情報を取得し、Ｓ４５においてその第２の描画情報中の論理演算コードが１６０であるか否かを判定する。論理演算コードが１６０であった場合、さらにＳ４６において、第２の描画情報と一緒に入力されてきた第２のパターン画像Ｐ’と座標情報を解析する。ここでの解析では、第２のパターン画像Ｐ’が白黒の２値画像であるか否かと、前に保持した描画情報との対象とする座標情報を判断する。解析の結果、第２のパターン画像Ｐ’が白黒２値画像であり、なおかつ対象とする座標が同一であると判断された場合、Ｓ４７において第２のパターン画像Ｐ’とその他の描画情報を描画オブジェクト記憶部２などに保存しておく。この場合も、この第２の描画情報に対するコマンドを出力せずに、第２の描画情報に関する処理を終了する。ここまでの処理で、２つの描画情報が描画オブジェクト記憶部２に保持されている。

【００５０】一方、Ｓ４４で取得した第２の描画情報中の論理演算コードが１６０でなかった場合と、Ｓ４６において第２のパターン画像Ｐ’が白黒２値画像ではないと判断された場合、あるいは、対象とする座標が異なった場合には、Ｓ５３において、描画オブジェクト記憶部２に保存された第１の描画情報からコマンドおよび描画オブジェクトを生成してプリンタ２０に送信する。次にＳ５４において、第２の描画情報に基づいてコマンドおよび描画オブジェクトを生成してプリンタ２０に送信する。この場合のコマンドは、通常の描画コマンドとしてプリンタ２０で処理される。

【００５１】次にＳ４８において、第３の描画情報を取得して、Ｓ４９においてその第３の描画情報中の論理演算コードが９０であるか否かを判定する。論理演算コードが９０であった場合、さらにＳ５０において、第３の描画情報と一緒に入力されてきた第３のパターン画像Ｐ”と座標情報を解析する。この解析では、この第３のパターン画像Ｐ”が、第１の描画情報に対応する第１のパターン画像Ｐと同一なものであるか否かと、対象とする座標が保持している第１の描画情報の座標と一致するかを判断する。解析の結果、第３のパターン画像Ｐ”が、第１のパターン画像Ｐと同一であり、なおかつ対象とする座標も一致すると判断された場合、ここまでの一連の第１ないし第３の描画情報は、マスキングパターン処理に変換可能であると判断する。そしてＳ５１において、第２の描画情報に対応する第２のパターン画像Ｐ’をマスキングパターンに変換するとともに、指定するパターン画像として第１または第３の描画情報に対応する第１のパターン画像Ｐまたは第３のパターン画像Ｐ”としたマスキングパターン処理を行う命令としてコマンドを生成し、プリンタ２０に対して送信する。この場合も、プリンタ２０における描画はマスキングパターンに従って、パターンイメージをディスティネーション画像に対して貼り付けてゆく処理であり、各画像データ間での論理演算処理は必要なくなる。

【００５２】一方、Ｓ４８で取得した第３の描画情報中の論理演算コードが９０でなかった場合と、Ｓ５０において、第３のパターン画像Ｐ”が第１のパターン画像Ｐと一致していなかった場合、あるいは対象とする座標情報が異なっていた場合には、第１ないし第３の描画情報それぞれについてコマンドおよび描画オブジェクトを生成して出力する。すなわち、Ｓ５５において、描画オブジェクト記憶部２に保存されている第１の描画情報からコマンドおよび描画オブジェクトを生成してプリンタ２０に送信する。次にＳ５６において、同じく描画オブジェクト記憶部２に保存されている第２の描画情報に基づいてコマンドおよび描画オブジェクトを生成してプリンタ２０に送信する。最後にＳ５７において、第３の描画情報に基づいてコマンドおよび描画オブジェクトを生成してプリンタ２０に送信する。この場合の各コマンドは、通常の描画コマンドとしてプリンタ２０で処理されることになる。

【００５３】上述の第１および第２の具体例では、それぞれ、描画論理演算ＰＳＤＰｘａｘ、あるいは、連続する３つの描画論理演算ＤＰｘ，ＤＰａ，ＤＰｘを検出した。これらは一例であって、他の論理演算処理を行わなくても処理可能な１ないし連続した複数の描画論理演算指示について、同様に論理演算処理を含まない指示および描画オブジェクトに変換することができる。

【００５４】また、上述のような第１および第２の具体例についての処理は単独で組み込むほか、組み合わせて組み込んでもよい。もちろん、他の論理演算処理を行わなくても処理可能な１ないし連続した複数の描画論理演算指示とともに判定し、処理してもよい。例えば、ある描画情報を取得したとき、その描画情報中の論理演算コードが１８４であるのか、９０であるのか、あるいはその他の変換対象の論理演算コードであるのかを判定し、論理演算コードが１８４であれば図５に示す処理を、論理演算コードが９０であれば図７に示す処理を行う等といったように、論理演算コードに応じた処理を行うように構成することが可能である。

【００５５】また上述の第１および第２の具体例では、オペレーティングシステム１４を介したアプリケーションプログラム１２からの描画情報を、特定の論理演算コードを持った描画情報についてプリンタドライバ１３の内部で解析を行い、画像データ間の論理演算を必要としないコマンドを生成した。しかしながら、このような解析とコマンドの変換処理は、プリンタドライバ１３の内部での処理に限定するものではない。すなわち、プリンタドライバ１３の内部では通常の描画コマンドとして処理を行い、その結果として、特定の論理演算を指定するコマンドもプリンタ２０へ送信する。そしてプリンタ２０側において、上述のような解析を例えばインタープリ

BNSDOCID: <JP____2000132671A__I_>

タ部21において実行することによって、結果として、論理演算を必要としない内部処理に変換することも可能である。この場合、図1に示した描画命令は、プリンタ20が受け取るコマンド、あるいはインタープリタ部21で生成した内部コードとすることができる。

【0056】以下、第3の具体例について説明する。この第3の具体例では、図1に示すような構成の本発明の画像処理方法をプリンタ20の内部で、インタプリタ部21に組み込んだ例について説明する。ユーザは、アプリケーションプログラム12において、例えばグラフィカルユーザインタフェース（GUI）を使用して任意に文字、グラフィックス、イメージ等の描画オブジェクトを画面上に配置していく。これらの描画オブジェクトは、重なりを持たせたり、半透明状態を指定するなど、種々の指定を行うことができる。

【0057】これらの描画方法のなかで、とくに、複数のグラフィックスデータを用いてグラデーション画像を作成することができる。図8は、論理演算を使用したグラデーション画像の生成方法の一例の説明図である。図8（A）は、グラデーションを描画する前のディスティネーション画像Dを示しており、三角形状の図形が描かれている例を表している。次に、図8（B）では、ソース画像S1からソース画像S5まで、SDxの演算、すなわちディスティネーション画像Dとソース画像Sの排他的論理和を行って描画する。次に、図8（C）では、値0の色（RGB色空間では黒）によって最終的にグラデーション部分を残したい形状で描画する。この例では、ソース画像S6として示す楕円形状の図形を描画している。最後に、図8（D）では、図8（B）と同様な描画を行う。すなわち、ソース画像S1からソース画像S5を使用して、SDxの演算を行って描画する。

【0058】ここで、排他的論理和は、同一の位置に同一の色で2回描画すると、処理を行う前の画素を保存する性質がある。また、値が0である画像に対しての排他的論理和は、単純に上書き描画した場合と同様な結果となる。図9は、排他的論理和の演算結果の説明図である。図9（A）は、ディスティネーション画像Dに対し、同一のソース画像Sで2回排他的論理和を行った場合を示している。すなわち、ディスティネーション画像Dとソース画像の排他的論理和はDSxであり、その結果に対してソース画像Sで排他的論理和を演算することはSDSxxで示される。図9（A）では、ディスティネーション画像Dとソース画像Sのすべての取りうる値の組み合わせを示している。図9（A）からわかるように、SDSxxの演算結果は、もとの画像であるディスティネーション画像Dと一致している。

【0059】また、図9（B）には、値が0のディスティネーション画像Dに対し、ソース画像Sと排他的論理和をとった場合を示している。このときの演算はDSxである。図9（B）に示すように、DSxの演算結果とソース画像Sは一致する。

【0060】この結果、図8（C）に示す画像に対してソース画像S1からソース画像S5までの画像で排他的論理和を取ることにより、値0の色によって描画した楕円形状の部分以外の部分では、図9（A）で説明したようにもとのディスティネーション画像Dが現れる。また、楕円形状の部分では図9（B）で説明したようにソース画像S1からソース画像S5までの画像が現出する。すなわち、図8（D）では、ちょうど図8（C）の処理で値が0の色で書き込んだ楕円形状の部分だけグラデーションがクリップされた結果として出力されることになる。このような描画方法によれば、どのような形状であっても自由にグラデーションを描画することが可能となる。

【0061】この一連の処理を表す描画命令をプリンタドライバ13がアプリケーションプログラム12から受け取って、通信路19を介してプリンタ20内部のインタプリタ部21に送る場合、一般的には、PDL（ページ記述言語）と呼ばれる描画命令群により送られる。図10は、ページ記述言語の一例の説明図、図11は、ページ記述言語において用いられる論理演算コードの一例の説明図である。ここでは、setattribute命令、drawrect命令、drawellipse命令、startclip命令、endclip命令について示している。

【0062】setattribute命令は、以降に描画される図形の属性を設定する。setattribute命令のパラメータ数は4であり、1から3番目のパラメータは、RGB色空間での色値をR，G，Bの順番で表す。4番目のパラメータは、論理演算のコードを表している。論理演算と論理演算のコードとの対応は、図11に示す通りであり、1から16までの16種類が定義されている。

【0063】また、drawrect命令は、矩形形状の図形を、現在の属性に従って塗りつぶして描画する。drawrect命令のパラメータ数は4であり、2つの対角点のx、及びy座標を表す。また、drawellipse命令は、楕円形状の図形を、現在の属性に従って塗りつぶして描画する。drawellipse命令のパラメータ数は4であり、楕円形状を取り囲む外接矩形の2つの対角点のx、及びy座標を表す。

【0064】また、startclip命令、endclip命令は、クリップ領域を作成するための命令であり、パラメータを必要としない。この2つの命令で囲まれたステートメント内部で発行された図形描画命令は、クリップ領域へと変換される。クリップ処理とは、ある図形を任意の形に切り取る処理であり、例えば特開平6-274643号公報に示される手法などによって実現することができる。

【0065】図12は、第3の具体例を描画するための

BNSDOCID: <JP_____2000132671A_I_>

描画命令群の一例の説明図である。図１２（Ａ）は、図１０，図１１で説明したように定義されるＰＤＬを用いて、図８に示したグラデーション画像の描画を記述した場合を示している。１０行目までのｓｅｔａｔｔｒｉｂｕｔｅ命令とｄｒａｗｒｅｃｔ命令は、図８（Ｂ）の描画に対応している。すなわち、２つの命令ごとにソース画像Ｓ１からソース画像Ｓ５を描画している。次の１１及び１２行目に記述されている命令は、ＲＧＢ空間での色をＲ，Ｇ，Ｂ＝０，０，０とした楕円形状を描画する命令である。この描画は、図８（Ｃ）に示した楕円形状の描画に対応している。さらに１３行目から最終行までの命令は、ソース画像Ｓ１からソース画像Ｓ５を描画する命令であり、図８（Ｄ）における描画に対応する。

【００６６】これらの命令群は、図８（Ｃ）において描画した図形形状（この例では楕円形状）が、最終的に残したいグラデーション図形の形状となる。すなわち、グラデーション図形を、図８（Ｃ）において描画した図形形状でクリップ処理した場合と等価である。これを利用し、ｓｔａｒｔｃｌｉｐ命令、ｅｎｄｃｌｉｐを用いると、図１２（Ａ）に示したＰＤＬは、図１２（Ｂ）のように書き換えることが可能となる。図１２（Ｂ）に示す記述では、はじめに１～３行目において、図８（Ｃ）における楕円形状の描画部分をクリップ領域として設定する。その後に、４～１３行目において、一連のグラデーション図形を描画する。この４～１３行目の記述は、図１２（Ａ）における１～１０行目あるいは１３～２２行目と同じであるが、論理演算コードを１３番（ソース画像による上書き）に変更している。

【００６７】このように、図１２（Ａ）では、論理演算として排他的論理和を使用して図８に示したように描画したが、図１２（Ｂ）ではクリップ命令を用いることによって、全て、通常の上書き描画で表現できることがわかる。本発明では、値が０の図形の描画を挟んで同じソース画像を排他的論理和により描画する命令が並んでいる場合、クリップ命令を用いた描画命令列に変換することによって論理演算を行わない描画命令列に変換する。これによって描画処理を高速化することが可能になる。また、論理演算を行わないことによって、画質の劣化を回避することができる。

【００６８】図１３ないし図１５は、本発明の実施の一形態における第３の具体的な変換処理の一例を示すフローチャートである。図１３および図１４で示す処理は、内部状態としてａからｄの状態遷移を行う。状態ａは初期状態を表す。状態ｂは、例えば図８（Ｂ）に示すソース画像Ｓ１からソース画像Ｓ５に対応する描画命令を受け取っている状態、つまり、図１２（Ａ）の１０行目までの命令とパラメータのセットを受け取る状態を表す。また、この状態ｂで受け取る命令とパラメータのセットの数をＣＯＵＮＴ１変数に記憶する。状態ｃは、図８（Ｃ）に示す値が０の色の楕円形状を描画する描画命令を受け取っている状態、つまり、図１２（Ａ）の１１，１２行目の命令とパラメータのセットを受け取っている状態を表す。また、この状態ｃで受け取る命令とパラメータのセットをＣＯＵＮＴ２変数に記憶する。状態ｄは、図８（Ｄ）に示すソース画像Ｓ１からソース画像Ｓ５に対応する描画命令を受け取っている状態、つまり、図１２（Ａ）の１３行目以降を受け取っている状態を表す。また、この状態ｄで受け取る命令とパラメータのセットをＣＯＵＮＴ３変数に記憶する。さらに、これらＣＯＵＮＴ１からＣＯＵＮＴ３の各変数に対応した命令とパラメータは、それぞれバッファ１からバッファ３へと格納されるものとする。

【００６９】まず処理開始直後のＳ６１において、内部状態をａに初期化するとともに、ＣＯＵＮＴ１からＣＯＵＮＴ３までの各変数を０に初期化する。次のＳ６２において、ＰＤＬの読み込みを継続するか否かを判定する。これは、ＰＤＬが印刷命令を発行していたり、何らかの理由で処理継続を中断する場合の判定を行う。読み込みを継続する場合には、Ｓ６３において、コマンドおよびパラメータのセットを１つ読み込む。

【００７０】Ｓ６４において、読み込んだコマンドおよびパラメータがｓｅｔａｔｔｒｉｂｕｔｅ命令であるか否かを判定する。ｓｅｔａｔｔｒｉｂｕｔｅ命令である場合には、Ｓ６５において、４番目のパラメータである論理演算コードが７番、すなわちＤＳｘ演算を指定しているか否かを判定する。ＤＳｘ演算を指定している場合には、さらにＳ６６において、状態がａであるか否かを判定し、状態がａであれば新たにＤＳｘ演算の命令列が始まったものとして、Ｓ６７において状態をｂに変更する。またＳ６８において、状態がｃであるか否かを判定し、状態がｃであれば値が０の色の図形を描画した後のＤＳｘ演算の命令列が始まったものとして、Ｓ６９において状態をｄに変更する。状態がそれ以外の場合には、状態の変更を行わずにＳ７５へ進む。

【００７１】ｓｅｔａｔｔｒｉｂｕｔｅ命令の４番目のパラメータが７番ではない場合にはＳ７０に進み、同じく４番目のパラメータが１３番、すなわちソース画像Ｓとして上書きを指定しているか否かを判定する。上書きを指定している場合、Ｓ７１において状態がｂで、色の値が０か否かを判定する。この場合、ＤＳｘ演算の描画命令列の後にクリップすべき図形を描画している可能性がある。そのため、Ｓ７２において状態をｃに変更してＳ７５に進む。また、Ｓ７３において状態がｃで、色の値が０の場合には、クリップすべき図形の描画が続いているものとして、状態を変更せずにＳ７５へ進む。これ以外の状態、あるいは色の値が０以外の場合には、クリップ命令への変換を行うことができないので、Ｓ８７へ進んでそれまでのＰＤＬをそのまま出力する。また、ｓｅｔａｔｔｒｉｂｕｔｅ命令の４番目のパラメータが７番でも１３番でもない論理演算が指定されている場合

BNSDOCID: <JP____2000132671A__I_>

も、クリップ命令への変換の対象外であるので、S87へ進んでそれまでのPDLをそのまま出力する。

【0072】S63で読み込んだ命令がsetattribute命令ではない場合、S74において、図形形状の描画命令であるか否かを判定する。drawrect命令やdrawellipse命令などのような図形形状の描画命令である場合には、命令をバッファにそのまま格納すべく、S75へ進む。setattribute命令でも、図形形状の描画命令でもない命令が現れた場合には、クリップ命令への変換を行わず、S87へ進んでそれまでのPDLをそのまま出力する。

【0073】S63で読み込んだ命令がsetattribute命令であり、DSx演算(7番)を指定している場合、あるいは、上書き(13番)を指定するとともに色の値が0でかつ状態がb、cの場合と、図形形状の描画命令である場合には、S75～S83において、命令をバッファに格納する。すなわち、S75において状態がbであると判定される場合には、S76においてCOUNT1変数を1増加させ、S77において命令をバッファ1に格納する。同様に、S78において状態がcであると判定される場合には、S79においてCOUNT2変数を1増加させ、S80において命令をバッファ2に格納する。S77，S80において命令をバッファ1またはバッファ2に格納した後、S62へ戻り、読み込みの継続を判定した後、次の命令についての処理を進める。

【0074】さらに、S81において状態がdであると判定される場合には、S82においてCOUNT3変数を1増加させ、S83において命令をバッファ3に格納する。この場合には、S84においてCOUNT3変数の値とCOUNT1変数の値を比較し、COUNT3変数の値がCOUNT1変数の値よりも小さい、すなわち、値が0の色の図形の描画の前に存在していたDSx演算の命令数よりも少ない場合には、S62へ戻って次の命令の処理を行う。COUNT3変数の値がCOUNT1変数の値が等しくなると、これをS85で検出し、クリップ命令による置き換えが可能な命令列を検出したものとして、S86において実際に置き換えが可能か否かを判定し、可能であれば置き換え処理を行う。S86の処理後は、新たな置き換え可能な命令列を検出すべく、S61へ戻る。

【0075】なお、COUNT3変数の値がCOUNT1変数の値よりも大きくなってしまったときには、クリップ命令への変換を行うことができないので、S87へ進んでそれまでのPDLをそのまま出力する。また、状態がaの場合にも、バッファに格納することなく、S87においてそれまでのPDLをそのまま出力する。S87においてPDLを出力した後は、S61に戻り、新たな置き換え可能な命令列を検出すべく処理を繰り返す。

【0076】最後に、読み込むべき描画命令が無くなるなど、読み込みを継続しない場合には、S88においてそれまでのPDLをそのまま出力し、処理を終了する。

【0077】S86において行う、実際に置き換えが可能か否かを判定し、可能であれば置き換え処理を行う処理を図15に示している。S91において、バッファ1とバッファ3の内容を比較する。この場合、バッファ1およびバッファ3に格納されている命令の数はCOUNT1変数あるいはCOUNT3変数により示されている。S92において、S91での比較の結果、バッファ1の内容とバッファ3の内容が一致しているか否かを判定する。一致している場合には、同じソース画像を排他的論理和で2回論理演算を行っていることを示している。また、図13，図14の処理において、これらの排他的論理和の命令列の間に、値が0の色による図形の描画が行われていることもわかっている。そのため、検出した命令列はクリップ命令を用いて書き換えが可能であると判断することができる。

【0078】この判断を受け、S93において、クリップ命令を用いた描画命令列に変換する。すなわち、startclip命令を出力し、バッファ2に格納されている図形形状の描画命令(コマンド/パラメータ)を出力し、endclip命令を出力する。さらに、バッファ1に格納されているすべての命令(コマンド/パラメータ)に対して論理演算コードを13番(ソース画像として上書きする演算)に変更して出力する。このようにして、論理演算を用いない描画命令列に変換することができる。

【0079】なお、S92において、バッファ1の内容とバッファ2の内容が一致しない場合には、同じソース画像を2回排他的論理和により演算しているとは言えないので、変換処理を行わずに、S94において蓄積されているすべてのPDLをそのまま出力する。

【0080】上述の変換処理を、図8で示した具体例を描画する描画命令列、すなわち図12(A)に示すPDLを図12(B)に示すPDLに変換する場合について説明する。図12(A)では、まず1行目の「setattribute 30 30 30 7」を読み込むことになる。この場合、命令はsetattribute命令であり、4番目のパラメータはDSx演算(7番)である。また、初期状態はaであるので、S67において、状態をbに変更する。状態がbに変更されているので、S76においてCOUNT1変数に1を加え、S77においてこの1行目の命令をバッファ1に格納する。そしてS62へ戻る。

【0081】次の命令、つまり図12(A)における2行目の「drawrect 5010 70 90」を受け取る。この場合、setattribute命令ではないが、図形形状の描画命令である。また、状態はbであるので、S76においてCOUNT1変数に1を加え、S77において2行目の命令をバッファ1に格納す

BNSDOCID: <JP_____2000132671A_I_>

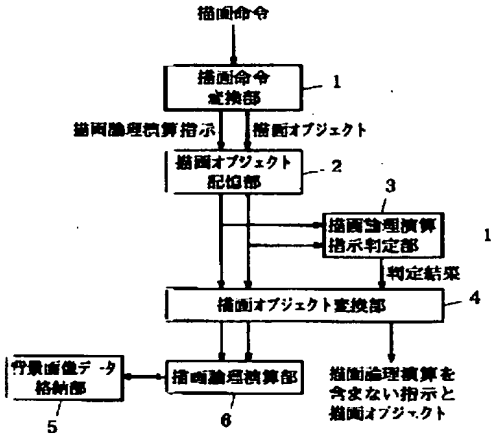
る。そしてＳ６２へ戻る。
【００８２】次の３行目の命令は、ｓｅｔａｔｔｒｉｂｕｔｅ命令であり、４番目のパラメータはＤＳｘ演算（７番）であるが、状態はｂであるので状態の変更を行わず、Ｓ７６においてＣＯＵＮＴ１変数に１を加え、Ｓ７７においてこの３行目の命令をバッファ１に格納する。そしてＳ６２へ戻る。
【００８３】このようにして、図１２（Ａ）の１０行目までは、この２つの流れを交互に繰り返すことになる。これにより、バッファ１に１０行目までの描画命令が格納される。
【００８４】次に、１１行目の「ｓｅｔａｔｔｒｉｂｕｔｅ　０　０　０　１３」を読み込むと、ｓｅｔａｔｔｒｉｂｕｔｅ命令ではあるが、４番目のパラメータはＤＳｘ演算（７番）ではない。しかし、４番目のパラメータはソース画像の上書きを示す１３番の論理演算コードであるので、さらに状態と色を判定する。この命令の時点では状態はｂであり、この命令の描画色の値はＲ，Ｇ，Ｂ＝０，０，０である。そのため、Ｓ７２において状態をｃに変更する。状態がｃに変更されているので、Ｓ７９においてＣＯＵＮＴ２変数に１を加え、Ｓ８０においてこの１１行目の命令をバッファ２に格納する。そしてＳ６２へ戻る。次の１２行目のｄｒａｗｅｌｌｉｐｓｅ命令は、図形形状を描画する命令であるので、Ｓ７９においてＣＯＵＮＴ２変数に１を加え、Ｓ８０においてバッファ２に格納される。
【００８５】次に、１３行目の「ｓｅｔａｔｔｒｉｂｕｔｅ　３０　３０　３０　７」を読み込むと、命令がｓｅｔａｔｔｒｉｂｕｔｅ命令であり、４番目のパラメータはＤＳｘ演算（７番）であり、さらに状態がｃであるので、Ｓ６９において状態をｄに変更する。状態がｄに変更されているので、Ｓ８２においてＣＯＵＮＴ３変数に１を加え、Ｓ８３においてこの１３行目の命令をバッファ３に格納する。そしてＳ６２へ戻る。以下、１４行目から２２行目の命令についても、Ｓ８２においてＣＯＵＮＴ３変数に１を加え、Ｓ８３においてバッファ３に格納される。
【００８６】２２行目の命令までを処理した時点で、ＣＯＵＮＴ１＝１０、ＣＯＵＮＴ２＝２、ＣＯＵＮＴ３＝１０となっている。ＣＯＵＮＴ１変数の値とＣＯＵＮＴ３変数の値が等しいので、図１５に示す処理を実行する。
【００８７】図１２（Ａ）に示す描画命令列では、１～１０行目の命令列と、１３～２２行目の命令列は等しいので、クリップ処理を用いた命令列に変換できると判断される。そして、Ｓ９３において、次の処理を行う。
（Ａ）ｓｔａｒｔｃｌｉｐ命令を出力する。
（Ｂ）バッファ２の図形描画の命令とパラメータを全て出力する。
（Ｃ）ｅｎｄｃｌｉｐ命令を出力する。
（Ｄ）バッファ１の命令とパラメータを全て出力する。この時、ｓｅｔａｔｔｒｉｂｕｔｅ命令のパラメータで論理演算を指定する４番目のパラメータには１３番（ソース画像の上書き描画）に変更して出力する。
以上の（Ａ）から（Ｄ）の４つの処理を実行することで、図１２（Ａ）に示される論理演算を使用した一連のグラデーション描画命令は、図１２（Ｂ）に示すように置換される。このようにして、論理演算を含む図１２（Ａ）に示す描画命令列を、論理演算を含まない図１２（Ｂ）に示す描画命令列に変換することができる。これにより、描画処理速度を向上させることができる。
【００８８】なお、この第３の具体例では、図１におけるプリンタ２０の内部で変換処理を行うものとして説明した。しかしこれに限らず、上述の第１，第２の具体例と同様に、プリンタドライバ１３で上述のような変換処理を行ってもよい。また、この第３の具体例も、上述の第１および第２の具体例とともに、組み合わせて実行できるように構成することも可能である。
【００８９】また、上述の第１ないし第３の具体例では、特に図２に示したような画像形成システムをもとに説明したが、本発明はこれに限らず、画像を形成する装置を含む任意のシステム構成において適用可能である。
【００９０】
【発明の効果】以上の説明から明らかなように、本発明によれば、論理演算処理を含む描画命令について、指示された論理演算処理をそのまま実行することが必要でない場合に、論理演算処理を含まないコマンドに変換して実行することができるので、論理演算処理にかかる時間を省いた高速な画像処理が可能となる。さらに、論理演算処理を行わないことによって、論理演算処理に伴う色変換処理の結果生じる画質上の不具合を回避することができるので、結果として高速な処理と高画質を両立した画像処理が実現できるという効果がある。
【図面の簡単な説明】
【図１】　本発明の実施の一形態を示すブロック図である。
【図２】　本発明の実施の一形態を実現する画像形成システムの一例を示すブロック図である。
【図３】　描画論理演算ＰＳＤＰｘａｘの一例の説明図である。
【図４】　マスキングパターン処理の一例の説明図である。
【図５】　本発明の実施の一形態における第１の具体的な変換処理の一例を示すフローチャートである。
【図６】　複数の描画論理演算の組み合わせの一例の説明図である。
【図７】　本発明の実施の一形態における第２の具体的な変換処理の一例を示すフローチャートである。
【図８】　論理演算を使用したグラデーション画像の生成方法の一例の説明図である。

BNSDOCID: <JP____2000132671A__I_>

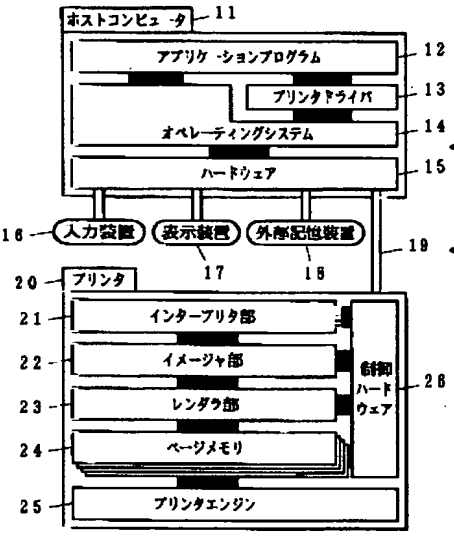
【図9】　排他的論理和の演算結果の説明図である。
【図10】　ページ記述言語の一例の説明図である。
【図11】　ページ記述言語において用いられる論理演算コードの一例の説明図である。
【図12】　第3の具体例を描画するための描画命令群の一例の説明図である。
【図13】　本発明の実施の一形態における第3の具体的な変換処理の一例を示すフローチャートである。
【図14】　本発明の実施の一形態における第3の具体的な変換処理の一例を示すフローチャート（続き）である。
【図15】　本発明の実施の一形態における第3の具体的な変換処理の一例を示すフローチャート（サブルーチン）である。

【符号の説明】
1…描画命令変換部、2…描画オブジェクト記憶部、3…描画論理演算指示判定部、4…描画オブジェクト変換部、5…背景画像データ格納部、6…描画論理演算部、11…ホストコンピュータ、12…アプリケーションプログラム、13…プリンタドライバ、14…オペレーティングシステム、15…ハードウェア、16…入力装置、17…表示装置、18…外部記憶装置、19…伝送路、20…プリンタ、21…インタープリタ部、22…イメージャ部、23…レンダラ部、24…ページメモリ、25…プリンタエンジン、26…制御ハードウェア。

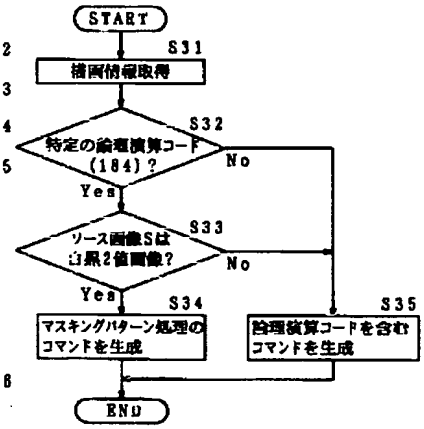
【図1】



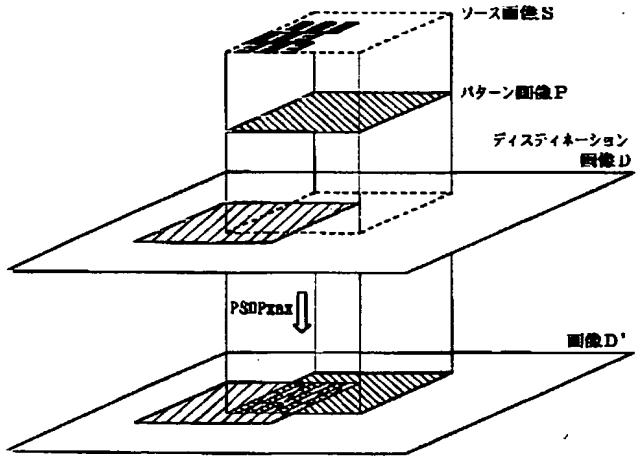
【図2】



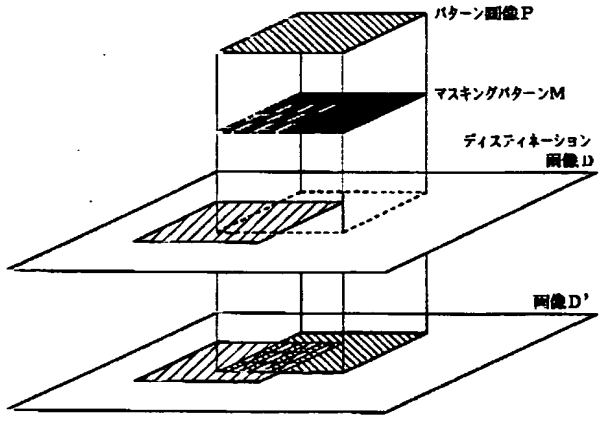
【図5】



【図3】

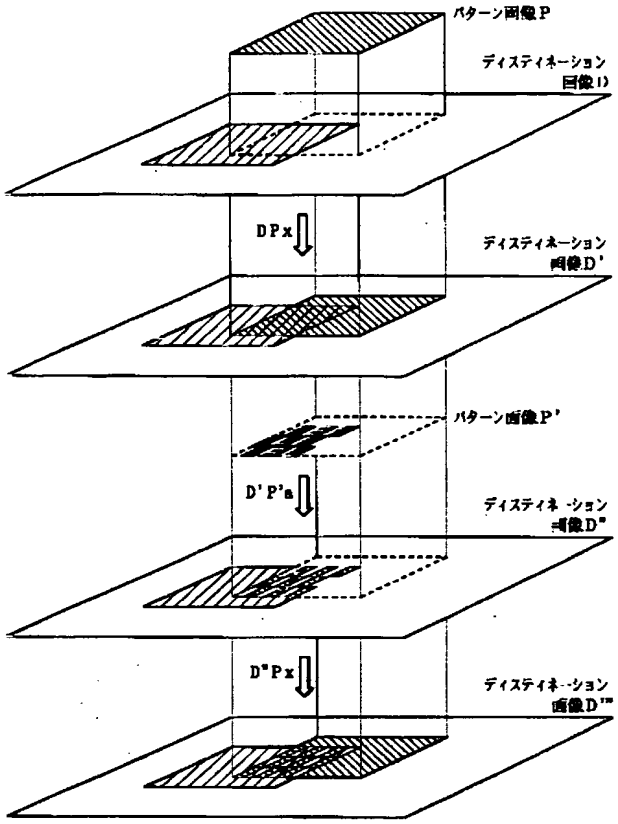


【図4】

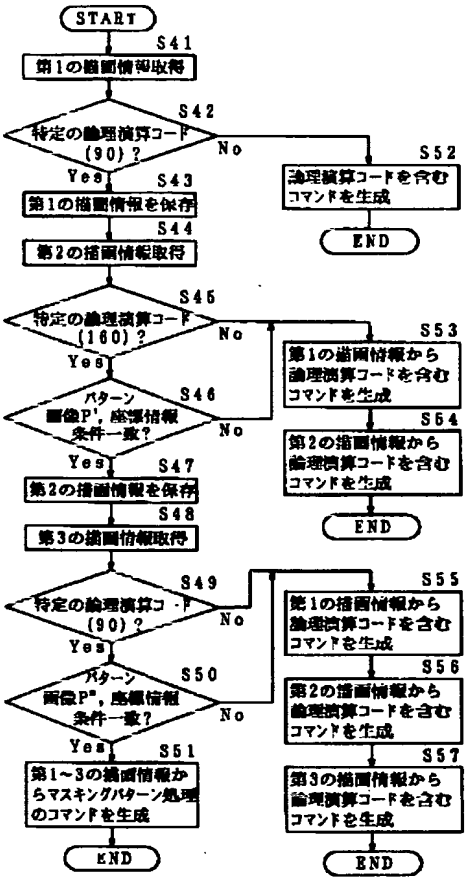


BNSDOCID: <JP_____2000132671A_I_>

【図6】



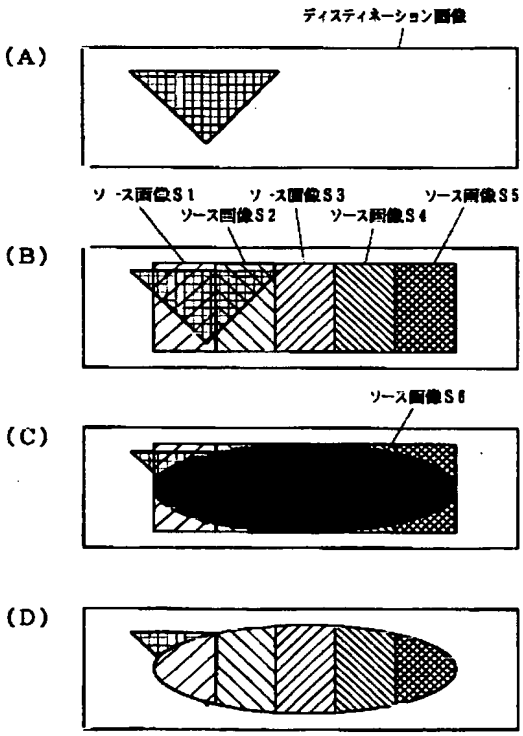
【図7】



【図11】

論理演算	コード
0	1
DSon	2
DSna	3
DSn	4
DSna	5
Dn	6
DSx	7
DSan	8
DSa	9
DSxn	10
D	11
DSno	12
S	13
SDno	14
DSo	15
1	16

【図8】



【図9】

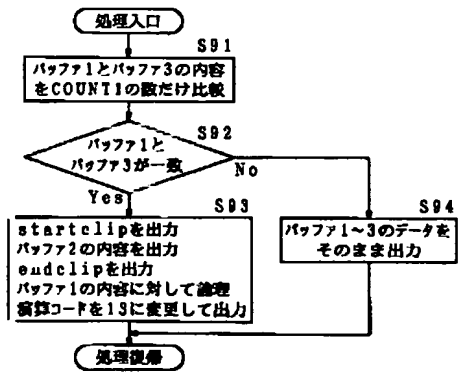
(A)

D	1	1	0	0
S	1	0	1	0
DSx	0	1	1	0
SDSxx	1	1	0	0

(B)

D	0	0
S	1	0
DSx	1	0

【図15】



BNSDOCID: <JP_____2000132671A__I_>

【図10】

コマンド名	パラメータ	
setattribute	c1	RGB色空間のRの値
	c2	RGB色空間のGの値
	c3	RGB色空間のBの値
	code	論理演算コード
drawrect	x1	対角点1のx座標
	y1	対角点1のy座標
	x2	対角点2のx座標
	y2	対角点2のy座標
drawellipse	x1	外接矩形の対角点1のx座標
	y1	外接矩形の対角点1のy座標
	x2	外接矩形の対角点2のx座標
	y2	外接矩形の対角点2のy座標
startclip	なし	
endclip	なし	

【図12】

(A)

```
 1  setattribute  30 30 30 7
 2  drawrect      50 10 70 90
 3  setattribute  60 60 60 7
 4  drawrect      70 10 90 90
 5  setattribute  90 90 90 7
 6  drawrect      90 10 110 90
 7  setattribute  120 120 120 7
 8  drawrect      110 10 130 90
 9  setattribute  160 160 160 7
10  drawrect      130 10 150 90
11  setattribute  0 0 0 13
12  drawellipse   60 20 140 80
13  setattribute  30 30 30 7
14  drawrect      50 10 70 90
15  setattribute  60 60 60 7
16  drawrect      70 10 90 90
17  setattribute  90 90 90 7
18  drawrect      90 10 110 90
19  setattribute  120 120 120 7
20  drawrect      110 10 130 90
21  setattribute  160 160 160 7
22  drawrect      130 10 150 90
```

グラデーション図形 排他的論理和演算

値が0の色の図形描画（楕円形状を上書き）

グラデーション図形 排他的論理和演算

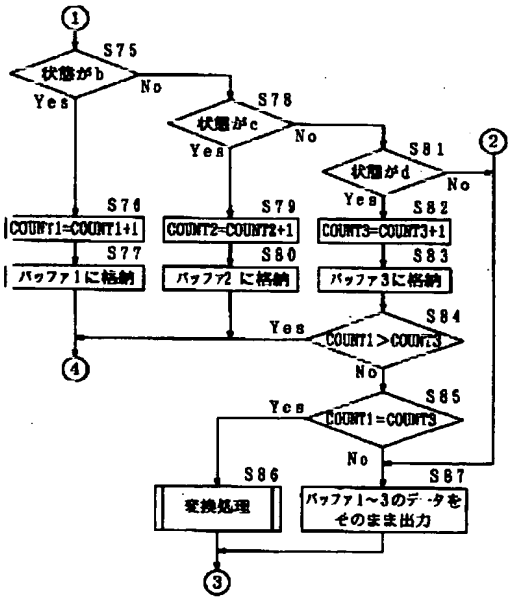
(B)

```
 1  startclip
 2  drawellipse   60 20 140 80
 3  endclip
 4  setattribute  30 30 30 13
 5  drawrect      50 10 70 90
 6  setattribute  60 60 60 13
 7  drawrect      70 10 90 90
 8  setattribute  90 90 90 13
 9  drawrect      90 10 110 90
10  setattribute  120 120 120 13
11  drawrect      110 10 130 90
12  setattribute  160 160 160 13
13  drawrect      130 10 150 90
```

クリップ領域を指定（楕円形状を指定）

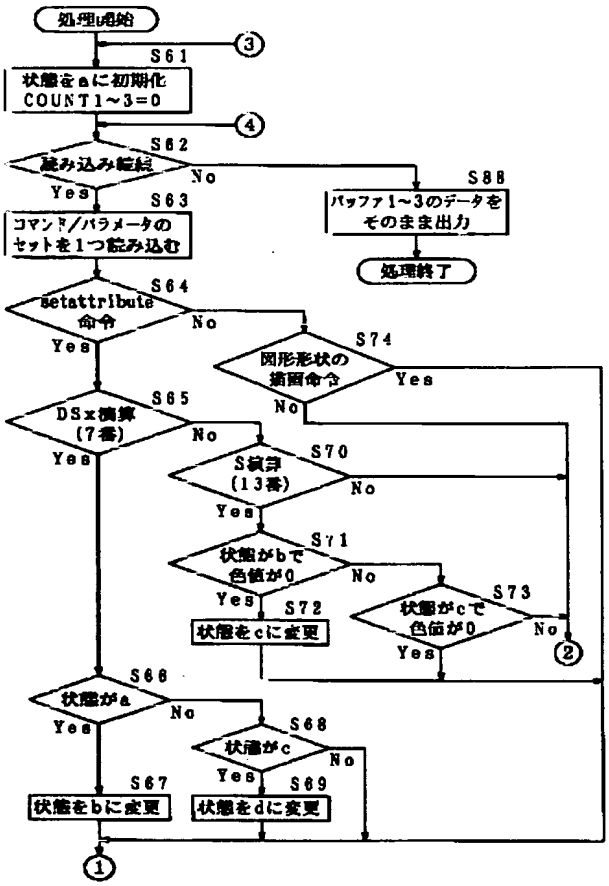
グラデーション図形 クリップ領域に上書き

【図14】



BNSDOCID: <JP_____2000132671A__I_>

【図13】



フロントページの続き

(72)発明者　小林　邦彦
神奈川県海老名市本郷2274番地　富士ゼロックス株式会社内
(72)発明者　関根　弘
神奈川県海老名市本郷2274番地　富士ゼロックス株式会社内

Fターム(参考)　5B057 CA18 CB01 CB16 CB18 CE18 CH01 CH11 CH18 DA08 DA17 DB06 DC25
5C076 AA02 AA26 AA40 BA05 BA06 BA10 CA12

BNSDOCID: <JP____2000132671A__I_>